REMARKS

Claims 1–37 are pending in the present application. In the Office Action mailed April 6, 2004, the Examiner rejected claims 1, 15–19, and 31-33 under 35 U.S.C. §103(a) as being unpatentable over Admitted Prior Art (APA) in view of Davidson et al. (USP 5,630,136). Claims 3–14 and 25–30 were objected to as being dependent upon a rejected base claim. Applicant appreciates the Examiner's indication that claims 34–36 are allowed.

In the Response mailed June 9, 2003, Applicant provided detailed remarks regarding the Examiner's rejection based on the APA and Davidson et al. In that Response Applicant provided a detailed analysis of the Examiner's failure to establish a *prima facie* case of obviousness. Applicant showed that (1) the combination of the APA and Davidson et al. did not teach or suggest a claimed method, (2) one of ordinary skill in the art would not have a reasonable expectation of success when combining the APA and Davidson et al. as the Examiner has done, and (3) the combination does not teach or suggest each and every element of the claims. Nonetheless, with respect to claims 1, 15–19, and 31–33, the Examiner ignored Applicant's remarks and has simply proffered the same rejection previously shown to be insufficient to establish a *prima facie* case of obviousness. Responsive to Applicant's remarks, the Examiner admitted that Davidson et al. "does not teach Java" but summarily concluded that because Davidson teaches the use of object-oriented environments, "one of ordinary skill in the art could apply the teachings of Davidson et al. in the Java language environment." Moreover, the Examiner has not provided any additional proof or support of this contention other that its assertion as fact. As such, the Examiner has fallen drastically short of that which is required to establish a *prima facie* case of obviousness.

The burden of establishing a *prima facie* case of obviousness falls squarely on the Examiner. MPEP §2142. Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent some teaching or suggestion supporting the combination. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). The MPEP states:

> To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success.

PAGE 12/17 * RCVD AT 5/24/2004 4:30:32 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:12623762994 * DURATION (mm-ss):04-26

Finally, the prior art reference (r references when combined) must teach or suggest all th claim limitations.

MPEP §2143

Applicant believes that a *prima facie* case of obviousness cannot be made based on the art of record because, as has been previously shown and reiterated below. Firstly, the references are directed to very different purposes and there is no motivation to combine these references in a way done so by the Examiner without the benefit of Applicant's own teaching. Secondly, the combination would not be successful. Thirdly, all the elements of the present claims are not present in the references. In short, the Examiner has not established the three basic criteria required under MPEP §2143.

Motivation to Combine

There is no motivation to combine Davidson et al. with the APA in the manner done so by the Examiner. Specifically, Davidson et al. teaches away from the use of Java. For example, Davidson et al. teaches "the baton processing 50 is executed on a computer and operating system that supports multithreading." Col. 5, ln. 66–col. 6, ln. 1 (emphasis added). One of ordinary skill in the art will readily recognize that such teaches directly away from the use of Java because Davidson et al. is explicit that the baton processing is executed within the operating system of the computer system on which it is being processed. On the other hand, the claimed invention is specifically directed to the use of Java which, as one of ordinary skill in the art will readily recognize, is processed within a Java virtual machine (JVM) and not within the operating system. Therefore, while Davidson et al. does disclose the use of object-oriented programming languages in general, and while Java is considered an object-oriented programming language, Davidson et al. is explicit that processing must occur within the operating system and therefore, necessarily precludes the use of Java. That is, were Java to be used within Davidson et al., processing would occur within the JVM rather than the operating system which is precluded by the teachings of Davidson et al.

Simply, one of ordinary skill in the art will readily recognize that programs written in most object oriented programming languages are typically designed to be executable within the operating system. However, programs written in Java, unlike other object-oriented programming languages, require that processing be handled by the JVM which exists on top of the operating system. That is, a hierarchy is created whereby the Java-based program executes within the JVM which exists within the operating system. On the other hand, Davidson et al. is explicit that processing be executed within the operating system. Col. 5, ln.

PAGE 13/17 * RCVD AT 5/24/2004 4:30:32 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:12623762994 * DURATION (mm-ss):04-26

66 – col. 6, ln. 1. Therefore, Davidson teaches directly away from the hierarchical system required in the claimed Java-based system. It is readily apparent, therefore, that Davidson et al. teaches directly away from the claimed invention because Davidson et al. precludes the use of programming languages such as Java that require processing outside of the operating system. The Examiner cannot simply ignore this express teaching by Davidson et al. and its preclusion of Java as a programming language.

Furthermore, even though the Examiner has failed to establish a *prima facie* case of obviousness, the fact that Davidson et al. teaches away from that which is claimed constitutes a rebuttal of a *prima facie* case of obviousness. MPEP §2144.05. That is, "[i]f the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *Id.* citing *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Modifying Davidson et al. to include Java would necessarily change the principle of operation because processing would occur within the JVM rather than within the operating system, as required by Davidson et al. Therefore, under MPEP §2144.05, the claims are not *prima facie* obvious.

Reasonable Expectation of Success

The combination of the APA and Davidson et al. would not be likely to succeed for the purpose of the claimed invention. That is, the claimed invention is directed to a system and method that "synchronizes a Java application and a preexisting XT intrinsic spaced visualization toolkit in a manner that prevents data corruption due to concurrency between the X event loop that services the tool kit and a thread or the threads or a Java application that make calls to the tool kit." Specification, pg. 5. The APA acknowledges, "Java does not directly provide any support for XT intrinsics and does not provide a method of supporting an X event loop." Specification, pg. 3. Therefore, the APA explains that prior art solutions "to using XT intrinsics based visualization toolkits within a Java application or Applet require that the application be separated into client and servo processes and have some form of interprocess communication." Specification, pg. 3–4. Similarly, Davidson et al. teaches "a distributed object-oriented environment (DOE) server 27 which is coupled between a client (model) 24 and an X server 28." Col. 5, lns. 40–43 (emphasis added). Therefore, Davidson et al. falls directly in the purview and explanation of the prior art as related in the APA. That is, the systems described in the APA and by Davidson et al. utilize a client-server

PAGE 14/17 * RCVD AT 5/24/2004 4:30:32 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:12623762994 * DURATION (mm-ss):04-26

configuration to achieve the desired synchronization between the application and the desired toolkit.

On the other hand, the claimed invention removes the need for a client/server configuration by providing a system and method that properly synchronizes the Java application and the preexisting software components in a manner that prevents data corruption due to concurrency between the software component and the portions of the Java application that call for the software. That is, the claimed system and method utilizes the inherent multithreading capabilities of Java to implement the necessary synchronization. On the other hand, as disclosed in the APA and as explicitly stated in Davidson et al., execution of these systems relies on the client/server relationship. Such is made expressly clear in Davidson et al. in stating that "[t]he baton processing 50 is executed on a computer system and operating system that supports multithreading." Col. 5, ln. 66–Col. 6, ln. 1. Simply, Davidson et al. teaches away from the use of threading within programming languages that natively support multithreading, such as Java, because Davidson et al. explicitly teaches that the operating system used must be a multithreading operating system. *Id.* Davidson et al. does not teach or suggest that any threads may be non-operating system threads, such as the claimed Java threads.

One of ordinary skill in the art looking to develop a system and method that synchronizes a Java application and preexisting XT intrinsic-based visualization toolkits in a manner that prevents data corruption due to concurrency between X event loops while not relying on a client-server configuration would not have a reasonable expectation of success in combining the ATA and Davidson et al. because (1) the combination is directed to the very client-server configurations being looked to overcome by the claimed invention, and (2) Davidson et al. teaches away from the use of multithreading within applications in favor or operating system multithreading. Simply, as explained in the APA and as disclosed in Davidson et al. these client-server solutions "require additional development time and effort to provide inner process communication and additional complexity needed to manage the client-server states." Specification, pg. 4. Therefore, one looking to create the claimed invention which overcomes these limitations inherent in client-server configurations would not be motivated to combine Davidson et al. and the APA because are both directed to the very client-server systems being looked to overcome. Furthermore, one of ordinary skill in the art would immediately recognize that the APA and Davidson et al., while both directed to

PAGE 15/17 * RCVD AT 5/24/2004 4:30:32 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:12623762994 * DURATION (mm-ss):04-26

client/server systems, are not compatible because the APA discusses prior art Java implementations while Davidson et al. teaches away from multithreading native programming languages such as Java.

Simply, the APA and Davidson et al. are the very systems which the claimed invention overcomes through the use of Java's multithreading capabilities as utilized in the specific manner claimed.

Elements of the Claims

The combination fails to teach or suggest each and every element of the claims as require under MPEP §2143 to establish a *prima facie* case of obviousness. Referring to claim 1, the Examiner asserted that the APA teaches "providing a JAVA application thread that includes a call to an X Window visualization toolkit (Java application requests a schedule camera position...is rotated), and a JAVA process thread that comprises an X Window X event loop (X event loop)." The Examiner cited page 4 of Applicant's Specification to support this assertion. However, the paragraph before the Examiner's citation clearly states, "[p]ast and present solutions to reusing Xt Intrinsics based visualization toolkits within a JAVA application or applet require that the application be separated into client and server processes and have some form of interprocess communication." Therefore, the APA does not teach "a JAVA process thread that comprises an X Window X event loop" but instead teaches separating the application into client and server processes. One of ordinary skill in the art will readily recognize that separating the application into client and server processes is not the same as dedicating a Java thread to an X event loop.

While this distinction alone renders the Examiner's rejection fatally flawed, Davidson et al. does not even teach the use of Java at all and, as described above, actually teaches away from Java. Nevertheless, the Examiner asserted, "[a]lthough Davidson does not teach a Java thread, Davidson teaches a thread in the object-oriented environment." As such, the Examiner concluded that "[i]t would have been obvious [that] the object oriented [environment] in Davidson could be Java environment." The Examiner's assertion is simply incorrect and inconsistent with Davidson et al. That is, as previously shown, Davidson clearly teaches that "[t]he baton processing 50 is executed on a computer system and operating system that supports multithreading." Col. 5, ln. 66–Col. 6, ln. 1. Therefore, it is readily apparent that Davidson et al. explicitly precludes the use of Java as an object-oriented

PAGE 16/17 * RCVD AT 5/24/2004 4:30:32 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:12623762994 * DURATION (mm-ss):04-26

programming language within its scope because, where Java to be used, such processing would be within the JVN and not the operating system as required by Davidson et al.

Additionally, claim 1 calls for a Java application thread. On the other hand, Davidson et al. teaches away from any application thread by expressly requiring baton processing within a multithreading capable operating system. Col. 5, ln. 66 – Col. 6, ln. 1.

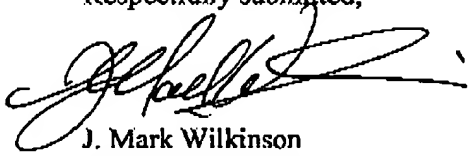
Therefore, Davidson et al. teaches directly away from the use of Java. As such, the combination does not teach each and every element of claim 1. Therefore, the Examiner has failed to establish a *prima facie* case of obviousness, and the rejection cannot be sustained.

With respect to claim 18, the Examiner again impermissibly concluded that Davidson et al. could be utilized in conjunction with Java and Java threads. However, as shown above, this conclusion is incorrect and unsupportable due to the fact that Davidson et al. teaches directly away from the use of Java or threads outside the operating system. As such, claim 18 is patentably distinct from the art of record.

Regarding claims 15–17, 19, and 31–33, Applicant believes the claims are in condition for allowance pursuant to the chain of dependency, as the claims depend from allowable claims 1 and 18. Therefore, in light of at least the foregoing, Applicant respectfully believes that the present application is in condition for allowance. As a result, Applicant respectfully requests timely issuance of a Notice of Allowance for claims 1–37.

Applicant appreciates the Examiner's consideration of these Remarks and cordially invites the Examiner to call the undersigned, should the Examiner consider any matters unresolved.

Respectfully submitted,



J. Mark Wilkinson
Registration No. 48,865
Direct Dial 262-376-5016

Dated: May 24, 2004
Attorney Docket No.: GEMS8081.029

P.O. ADDRESS:
Ziolkowski Patent Solutions Group, LLC
14135 North Cedarburg Road
Mequon, WI 53097-1416
262-376-5170

PAGE 17/17 * RCVD AT 5/24/2004 4:30:32 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:12623762994 * DURATION (mm-ss):04-26